KNOWLEDGE SWARMS: GENERATING EMERGENT SOCIAL STRUCTURE IN DYNAMIC ENVIRONMENTS

R.G. REYNOLDS,* B. PENG, and X. CHE,
Department of Computer Science, Wayne State University, Detroit, MI

ABSTRACT

Our previous work on real-valued function optimization problems had shown that cultural learning emerged as the result of meta-level interaction or swarming of knowledge sources (i.e., "knowledge swarms") in the belief space. These meta-level knowledge swarms induced the swarming of individuals in the population space (i.e., "cultural swarms"). The interaction of these knowledge swarms also produced emergent phases of problem solving at the population level that reflected an algorithmic process and resulted in the emergence of individual roles within the population: explorers and exploiters. Roles similar to this have been observed in animal populations and labeled "producers" and "scroungers," respectively (Barnard and Sibly 1981). Here we investigate the impact of environmental dynamics on the spatial and temporal aspects of role emergence. Specifically we generate a repeated shift in the resource landscapes at different intervals and note that this adds new distinctions within the previous role structure. That is, environmental complexity induces an increase in the complexity of social roles within a given system through the knowledge swarming process.

Keywords: Cultural algorithms, role emergence, cultural swarms, social intelligence, marginal value theorem

INTRODUCTION

Recently, a number of socially motivated algorithms have been used to solve optimization problems. Some of the example algorithms are the particle swarm algorithm (PSO) (Kennedy and Eberhart 1995), ant colony algorithm (ACO) (Dorigo et al. 1996), and cultural algorithm (CA) (Reynolds 1978, 1994). These three algorithms all use a population-based model as the backbone of the algorithm and solve problems by sharing information via social interaction among agents in the population.

Figure 1 expresses each of these approaches in terms of both a space and a time continuum over which the social interactions take place. Notice that both the ant and particle swarm approaches can be found near the lower left end of this continuum, with the social interaction between individuals taking place within limited temporal and spatial dimensions. For example, in particle swarm, agents can exchange their direction of movement and velocity locally with other agents. In the ant algorithm, agents locally exchange information in terms of the density and gradient of a "pheromone" substance that marks their trail. The pheromone

---

* *Corresponding author address*: Robert G. Reynolds, Department of Computer Science, Wayne State University, 409 State Hall, Detroit, MI 48202; e-mail: reynolds@cs.wayne.edu.

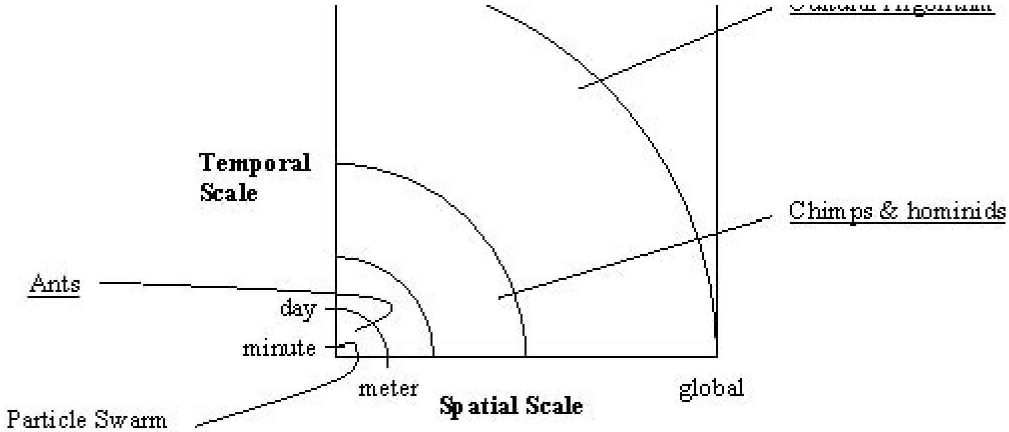

FIGURE 1 Scale of social interaction (The emergent properties depend on the scale at which the interaction takes place.)

chemical is deposited by an ant as it moves along a trail. The frequency of use of a trail is indicated by the amount of pheromone that is deposited relative to its degradation in the environment over time.

CAs, on the other hand, allow agents to interact in many different ways by using various forms of symbolic information reflective of complex cultural systems. The basic CA allows individuals to communicate via a shared belief space. The shared space stores five basic types of information that can be shared cognitively or symbolically.

Cultural Algorithms

A CA is a class of computational models derived from observing the cultural evolution process in nature (Reynolds 1978, 1994). A CA has three major components: a population space, a belief space, and a protocol that describes how knowledge is exchanged between the first two components. The population space can support any population-based computational model, such as genetic algorithms and evolutionary programming. The basic framework is shown in Figure 2.

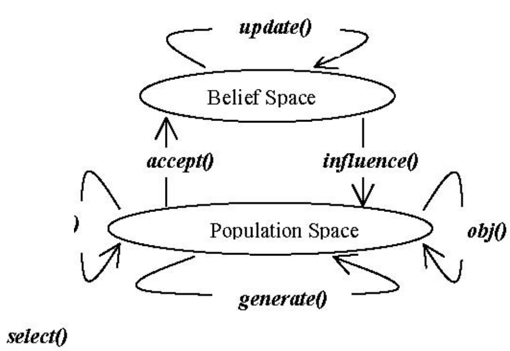

FIGURE 2 Framework of a cultural algorithm

A CA is a dual inheritance system that characterizes evolution in human culture at both the macro-evolutionary level, which occurs within the belief space, and the micro-evolutionary level, which occurs in the population space. Knowledge produced in the population space at the micro-evolutionary level is selectively accepted or passed to the belief space and used to adjust the knowledge structures there. This knowledge can then be used to influence the changes made by the population in the next generation.

What makes a CA different from the PSO and ACO approaches is the fact that a CA uses five basic knowledge models in the problem-solving process rather than just one or two locally transmitted values. There is ample evidence from the field of cognitive science that each of these knowledge models is supported by various animal species (Wynne 2001; Clayton et al. 2000), and it is assumed that human social systems support each of these models as well. The knowledge sources include normative knowledge (ranges of acceptable behaviors), situational knowledge (exemplars or memories of successful and unsuccessful solutions, etc.), domain knowledge (knowledge of domain objects, their relationships, and interactions), history knowledge (temporal patterns of behavior), and topographical knowledge (spatial patterns of behavior). This set of categories is viewed as being complete for a given domain in the sense that all available knowledge can be expressed in terms of a combination of one of these classifications.

Problem Statement

The CA has been studied with benchmark problems (Chung and Reynolds 1998) as well as applied successfully in a number of diverse application areas, such as modeling the evolution of agriculture (Reynolds 1986), concept learning (Reynolds 1994), real-valued function optimization (Jin 1999; Reynolds and Saleem 2005), re-engineering of semantic networks (Rychlyckyj 2003), and agent-based modeling of price incentive systems (Ostrowski and Reynolds 2002), among others.

While successful, the relative complexity of the knowledge sources and their interaction made it difficult to determine why CAs worked so well. Alternatively stated, under what conditions will such systems successfully solve a given problem, and what social structures will emerge along the way? The emergence of these structures in both the population and belief space can be viewed as signs of a successful problem-solving process.

In this paper, we attempt to develop answers to these questions. To do this, we begin by examining how CAs solve resource optimization problems within an experimental environment. In our investigation here, we employ a simulated cones world environment developed initially by Morrison and De Jong (1999) and extended here. Within this world, resources are viewed as being distributed in piles (cones) on the landscape (Sugarscape style; see Epstein and Axtell 1996).

In our paper, we use five different knowledge sources to direct the agents. Each knowledge source is a model for an agent's behavior. Since the belief space consists of five different knowledge sources or models, the question at each time-step is how to assign agents to the various models. The key here is that each knowledge source has an expression in two-dimensional (2D) space in terms of a bounding box characterized by a midpoint and standard deviation in the x and y directions. If we view each box as analogous to a resource patch in the environment, every knowledge source model can be viewed as a predator searching for prey in a given patch.

Since our cones world problems can then be described as foraging problems within a search space, we use a framework within the CA for the selection of a given knowledge source by an agent inspired by theoretical results from studies of foraging theory in population biology. Specifically, agents select different knowledge sources on the basis of what we characterize as "the marginal value of information." The inspiration for this comes from the classic work by Charnov (1976) concerning the "marginal value theorem." In certain situations, agents using the marginal value theorem were able to optimize their long-term resource intake within an environment. Simply stated, the marginal value theorem says that an agent stays within a location (patch) on the landscape until the current resource gain is less than the average expected value. It then moves to another patch.

Agents are then attracted to different knowledge sources on the basis of how successful individuals using each model are. In a previous paper (Reynolds and Peng 2005), we showed that this approach produced two classes of individuals — explorers and exploiters — depending on the particular knowledge models that they tended to use when the cones were configured in a static environment. As it turns out, these distinctions have been observed in naturally occurring animal populations as well. Barnard and Sibly (1981) identify "producers" who are engaged in finding resource patches and "scroungers" who exploit the found resources. Thus, our model was able to show how these roles might have emerged as a result of the knowledge swarming process within their shared belief space.

However, a question remains. What impact does the environment have on the emergence of these roles? In other words, do similar roles emerge in dynamically changing environments? In order to investigate this, we make the cone configurations in our model dynamic but predictable. A resource come is placed in each of four quadrants, and the cones are interchanged in counterclockwise fashion at regular intervals. We will show that each of the five knowledge sources or models adjusts it patch size and dynamics in a rather complementary fashion to exploit these dynamics. As a result, we see the emergence of subgroups of individuals within both the exploiter and explorer classes based on the models that they select to control their movement. In other words, the addition of environmental variability offered the agents more opportunities in terms of their knowledge models than were offered in the static case. This resulted in the production of a more complex social structure.

The next (second) main section describes the cones world environment. The third section describes the CA system configuration and how the marginal value theorem is employed here to adjust the patches for each knowledge source. The fourth section describes the simulation environment and experimental dynamics. The next section presents our results and describes some emergent properties of the social system, and the last section gives our conclusions.

THE CONES WORLD ENVIRONMENT

Our test problems are generated by a multi-modal problem generator DF1 (Morrison 1999), in which a "field of cones" of different heights and different slopes are randomly scattered across the landscape. The landscape is given by:

$$f\left(< x_1, x_2, ..., x_n >\right) = \max\left(H_j - R_j * \sqrt{\sum_{i=1}^{n}\left(x_i - C_{j,i}\right)^2} \right)$$

where $< x_1,\ldots,x_k >$ represent points in the landscape, n specifies the number of cones in the environment, and k is the number of dimensions. Each cone is independently specified by its location $< C_{j,1},\ldots,C_{j,n} >$, its height H_j, and its slope R_j. The cones are then "blended" together by using the max function to form the search surface.

The main reason that we picked this generator is that by changing its parameters, it can generate test functions over a wide range of surface complexity and problem dynamics. This enables us to evaluate our model in a more flexible and systematic way. An example 2D landscape is shown in Figure 3.

KNOWLEDGE SOURCES AND THE MARGINAL VALUE THEOREM

In this section, we briefly discuss the five knowledge sources used in the belief space and then motivate their integration in the optimization and search process by using the marginal value theorem. Each of the five knowledge sources or models has been observed to be cognized and used in various nonhuman species as a basis for encoding their social knowledge (Wynne 2001; Clayton et al. 2000).

The Five Knowledge Sources

For each knowledge type, we elaborate on its definition, the data structure, and how it is updated. Throughout the description, we use the symbol n for the number of parameters of the optimization problem. It is often referred to as dimensions of an optimization problem.

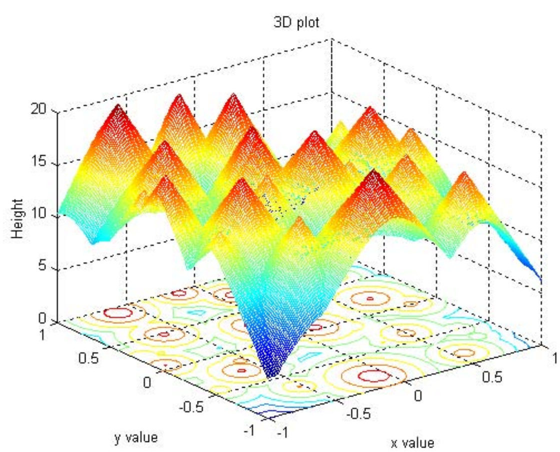

FIGURE 3 DF1 with n = 50, H ∈ (1, 10), R ∈ (8, 20), and in 2D space [(–1.0, 1.0), (–1.0, 1.0)]

Situational Knowledge

The situational knowledge source was first proposed by Chung (1997) for real-valued function optimization problem-solving in static environments. Situational knowledge contains a set of exemplars taken from the population. The data structure of the situational knowledge is represented as a list of exemplar individuals, as shown in Figure 4.

Each exemplar contains a value for each parameter and the fitness value for this exemplar. The situational knowledge will be updated either by adding the population's best individual to the situational knowledge if it outperforms the current best or reinitializing it when environmental change is detected. Situational knowledge represents exemplars or examples for other individuals to follow. These are case studies or events that are the basis for others behavior (Wynne 2001).

Normative Knowledge

Normative knowledge was also introduced by Chung (1997). It is represented as a set of intervals, and each is viewed to be a promising range for good or socially acceptable solutions for a parameter. The normative knowledge data structure for *n* variables is given as follows in Figure 5.

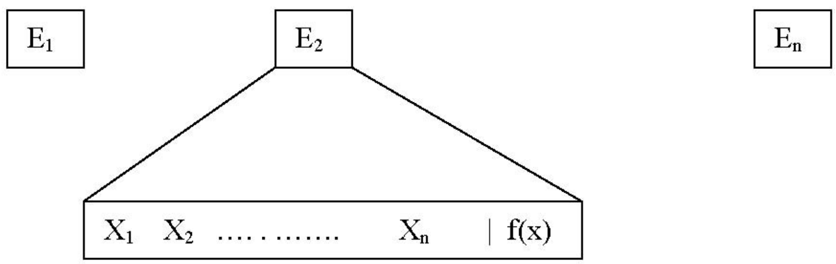

FIGURE 4 Structure of situational knowledge

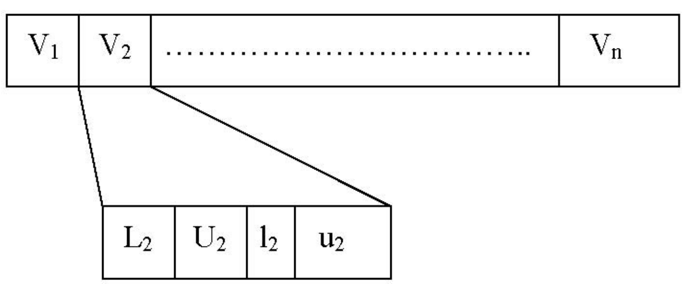

FIGURE 5 Structure of normative knowledge

For each variable V_i, the data structure contains the upper and the lower bounds, l_i and u_i, and the performance values for individuals in the upper and lower bounds, L_i and U_i. Normative knowledge is updated by shifting the variable ranges and updating the corresponding performance values to reflect changes in the environment.

Topographical Knowledge

Topographical knowledge, also called regional schemata (Jin 1999), is represented in terms of a multidimensional grid or array with cells in the grid described as $c_1,\ldots,c_i,\ldots,c_n$, where c_i is the cell size for the ith dimension. There is strong evidence for the ability of difference species to process 2D data displays. The data structure representation is an array of size n, where n is the number of cells in the mesh. Each cell in the data structure contains a lower and an upper bound for the n variables $[(l,u)1,\ldots,(l,u)n]$, indicating the ranges associated with the best solutions found in that cell so far, and a pointer to its children, as shown in Figure 6.

The topographical knowledge structure is initialized by the random placement of agents within cells in the grid and by creating a list of best cells. The update occurs when a cell is divided into subcells when an accepted individual's fitness value is better than the best solution in that cell, or if the fitness value of the cell's best solution has increased after a change event is detected. Topographical knowledge provides a spatial or array framework in which environmental patterns can be identified and adjusted for.

Domain Knowledge

Domain knowledge was introduced into the CA (Reynolds and Saleem 2005) in order to solve dynamic optimization problems. Domain knowledge was designed to reason about local dynamics. especially in terms of the prediction of gradients of incline or decline. Its data structure is shown in Figure 7.

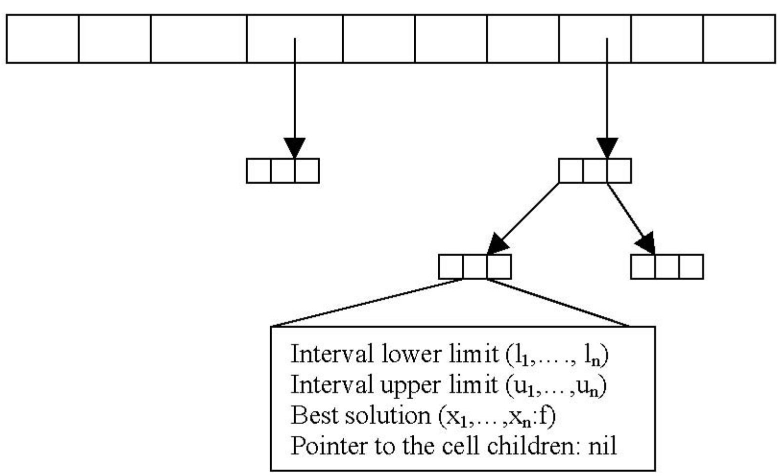

FIGURE 6 Structure of topographical knowledge

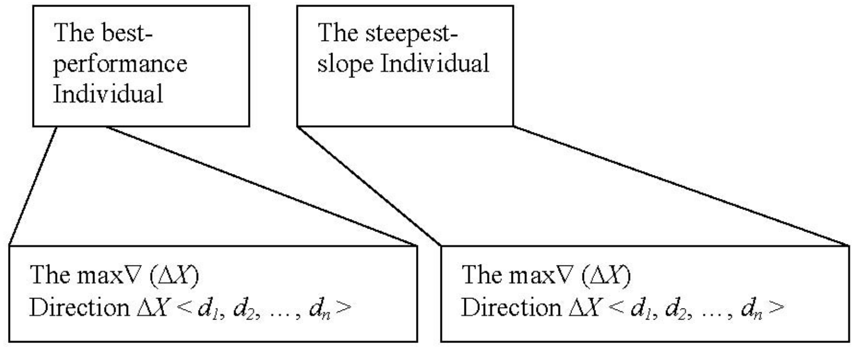

FIGURE 7 Structure of domain knowledge

Here domain knowledge consists of the domain ranges for all parameters and the best examples from the population space, similar to the situational knowledge representation above. Here domain knowledge is used to predict trends in the resource landscape both statically and dynamically. For example, given the cones world, if an upward slope or gradient is detected, then like an ant following a pheromone trial, one predicts a source increase. Likewise, domain knowledge can be used to predict the locations of resources in a dynamic environment. For example, if the amount of resources at a point is under the influence of a single cone, and if the slope at that point changes, then so has the point. One can predict the amount of shift necessary to place the slope at that point. This will allow agents using this model to make predictions about the future locations of a cone in the dynamic case.

History Knowledge

History knowledge was developed (Reynolds and Saleem 2005) in order to reason about global dynamics and to facilitate backtracking or retracing steps. It contains information about sequences of environmental changes in terms of shifts in the distance and direction of the known resource cones in the search space. Its cognitive origin comes from episodic memory (both in humans and animals), which is a type of memory based on personal experience. It stores information about events and temporal-spatial relations among those events (Clayton et al. 2000). While domain knowledge is focused on the interpretation of a resource shift locally in terms of geometrical or gradient considerations, history knowledge provides a more global perspective of the change. It computes the average change in parameter values within a region, the window size, and predicts the direction of the shift in the optimum from the previous position. The knowledge data structure representation is shown in Figure 8.

Here w represents the number of change events stored and $(ds_1,\dots,ds_n)$ and $(dr_1,\dots,dr_n)$ represent the average environmental changes in distance and direction, respectively, for each one of the n parameters. e_1 through e_w are change events. The history knowledge is updated after every change event by updating the history list and the moving averages for each parameter.

History knowledge is implemented as a list of up to m temporal events/points on the search path $\{P_1, P_2,\dots, P_m\}$. m is the size limit of the history list, and each $P_j = < p_j, 1, p_j, 2, \dots, p_j, n >$ represents a significant point on the search path.

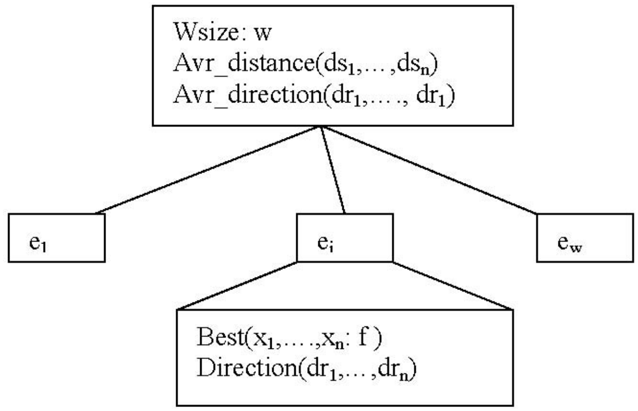

FIGURE 8 Structure of history knowledge

Communication Protocol

The communication protocol of a CA system is composed of two functions. The acceptance function determines which individuals are used to impact the belief space, and the influence function determines how the belief space influences the population space in generating a new solution.

Acceptance Function

The acceptance function determines which individuals and their behaviors can impact the belief space knowledge. It is often specified as a percentage of the number of current individuals, ranging between 1% and 100% of the population size, and based on selected parameters such as performance. For example, we can select the best performers (e.g., top 10%), worst performers (e.g., bottom 10%), or any combinations.

Influence Function: Using the Marginal Value Theorem

The choice of influence function has a great impact on the problem-solving process. Some influence functions are more successful than others, as measured by the success of the agents that each has influenced in the past. Early influence functions randomly applied the five knowledge sources to individuals in the population in order to guide their problem-solving process.

A good search approach should optimize the rate at which the available resources are processed by the foraging agents as they search for the optimum food search. While the distribution is continuous, it was observed that at each time-step that the individuals generated by each knowledge source using a normal distribution could be described by a “bounding box” or patch with a given central tendency and standard deviation. For example, in Figure 9, notice the shifting of the patch for situational knowledge from one location on the landscape to another. In fact, the original patch orientation is rotated and then translated toward the optimal point “+” over time.

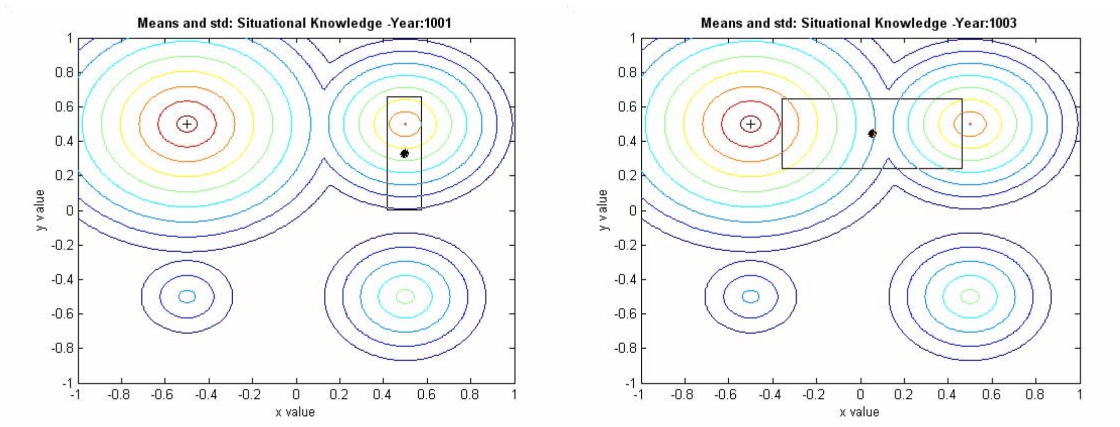

FIGURE 9 Situational means and standard deviation at year 1001 and 1003

In foraging theory, it had been shown that the use of the marginal value theorem is able, under certain conditions, to optimize the long-term average rate of energy intake within a patch-base environment (Charnov 1976). The principle behind the marginal value theorem is that residence time in a patch by a forager affects the expected energy gain. The marginal value principle states that the forager should reside in the patch "until the intake rate in a patch drops to the average rate for the habitat . . . it is the 'moving-on threshold' intake rate that is important" (Stephens and Krebs 1986, page 31). The forager, when doing so, will maximize the average long-term energy intake of the individual. One of the key assumptions is that the gain function associated with a patch is initially increasing but eventually negatively accelerated. Other assumptions are shown in Figure 10 taken from Stephens and Krebs (1986).

ASSUMPTIONS

Decision

The set of residence times for each patch type, t_i for patch type i.

Feasible choices: For all patch types $0 \leq t_i < \infty$.

Currency

Maximization of long-term average rate of energy intake.

Constraint

C.1 Searching for and hunting within patches are mutually exclusive activities.

C.2 Encounter with patches is sequential and is a Poisson process.

C.3 Encounter rates when searching are independent of the residence times chosen.

C.4 Net expected energy gain in a patch is related to residence time by a well-defined gain function [$g_i(t_i)$] with the following characteristics:

(i) Change in energy gain is zero when zero time is spent in a patch.

(ii) The function is initially increasing and eventually negatively accelerated.

C.5 Complete information is assumed. The forager knows the model's parameters and recognizes patch types, and it does not acquire and use information about patches while foraging in them.

FIGURE 10 Summary of the patch model (based on Stephens and Krebs 1986)

Figure 11 describes the calculation for a single patch. This figure is taken from Stephens and Krebs (1986, p. 30). There are two quantities plotted on the abscissa: travel time or placement effort and patch residence time. Each of the knowledge sources in the influence function is viewed as a predator. Travel time increases from the origin (vertical line) to the left, and patch residence time increases from the origin to the right. The gain function shape exhibits an initial increase and then escalating decrease. The optimal residence time can be found by constructing a line tangent to the gain function that begins at the point $1/\lambda$ on the travel time axis. The slope of this line is the long-term average rate of energy intake, because $1/\lambda$ is the average time required to travel between patches. When the travel time is long ($1/\lambda 2$), then the rate-maximizing residence time ($\hat{t}_2$) is long. When the travel time is short ($1/\lambda 1$), then the rate-maximizing residence time ($\hat{t}_1$) is shorter. Here travel time is a constant amount that represents a model time-step.

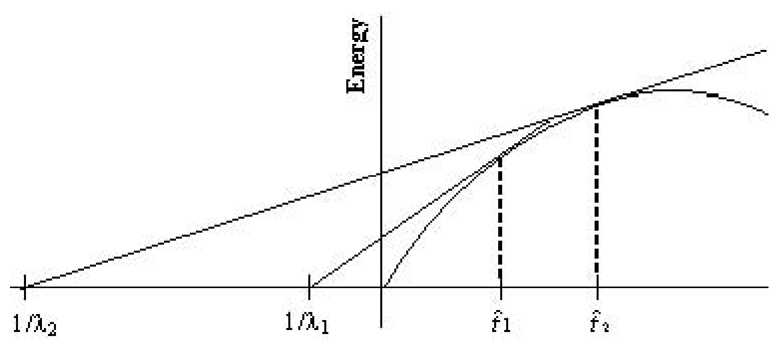

FIGURE 11 Marginal value theorem in the one-patch-type case

In this paper, we use an influence function based on the marginal value theorem as discussed in Peng (2005). The marginal value theorem is implemented here in terms of a roulette wheel function. The size of a knowledge source area under the wheel is a function of its ability to produce above-average gains. Each of the five knowledge sources, predators, is initially given 20% of the wheel area with which to generate its patch.

The likelihood of using one of the knowledge sources as a model for agent movement is based on the size of the area under the wheel, and the area for a knowledge source (predator) is adjusted on the basis of the performance of those agents it influences. At every time-step, each of the agents in the population is influenced by one of the knowledge sources on the basis of a spin of the wheel. The agent then moves to a position within the patch or bounding box associated with the selected knowledge source. The gain produced by the agent there is then recorded for the predator there.

The performance of a knowledge source can then be generated by computing the average fitness value of all individuals generated by each knowledge source. The average fitness value of individuals generated from using a specific knowledge source (predator) is:

$$avr_i = \frac{\sum_{j=1}^{k} f_j(x)}{k},$$

where k is the number of individuals generated via the knowledge source and is the fitness value of individual j.

Now each influence operator is assigned an area on the roulette wheel relative to its average performance, computed above, over the average performance for all of the influence functions:

$$p_i = \frac{avr_i}{\sum_{j=1}^{n} avr_j},$$

where p_i is a percentage on the roulette wheel assigned to influence operator i; and n is the number of influence operators used in the system.

When the value for a patch falls below the average, the area under the wheel will approach 0, and few individuals will be placed in that patch. However, its patch dimensions can be affected by the other active patches and new patch dimensions produced. If the patch shift is successful, the gain for the knowledge source will increase, and its share of the wheel will be become larger. At the same time, other knowledge sources will be experiencing a decrease in gain, and their areas will shrink.

Thus, with a gain function that increases initially and then decreases exponentially, we should get a phased pattern of knowledge use, where as some patches begin to fail, others are getting more individuals and increasing; however, with too much exploitation, they begin to fail and the cycle repeats itself. In the next section, we provide an example of how the bounding boxes for each of the knowledge sources (predators) shift during the course of the problem-solving process.

Experiment Settings

We set up a dynamic problem on the basis of the cones world problem described in Section 2. As a starting point, we construct a baseline landscape with four cones of different heights and slopes, where each is placed in one of four quadrants of the Cartesian plane (Figure 12). The cones can overlap on the basis of their slopes. Then the cones are shifted periodically 90° in the counterclockwise direction, so that every four 90° shifts form a complete rotation (Figure 12). We use this pattern to examine how the different knowledge sources react to this patterned movement and to observe the roles that emerge. Here, the cones are moved every 200 generations for a total of eight shifts or two cycles around after 1,600 generations.

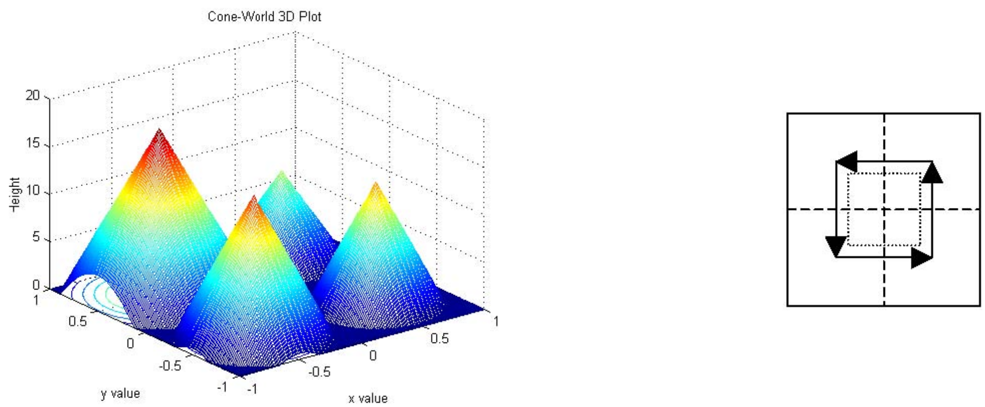

FIGURE 12 Dynamic cones world and the moving pattern

In all of our experiments, the parameters for CA were set as follows: the population size was set to 200, and the maximum number of generations between shifts was set to 200.

RESULTS AND DISCUSSION

We discuss the behavior of each knowledge source in terms of the placement of the individuals they influence in the search. The term “patch” is used here to indicate the area in which a given knowledge model is likely to place individuals. A patch is identified statistically by its center or the average location of those individuals that use it as a model this time-step, and its size is bounded by the standard deviation of the locations of these individuals. Figures 15 through 19 give the changing location of the center of the patch for each knowledge source over the 1,600 time-steps over the 2D grid in response to the movement of the cones. An arc connects the patch center at one time-step to the patch center at the next.

Here agents are given a number of knowledge models that they can use. Each time, the agents look at their previous performance and select the best model for them. From this perspective, we can see the strategy associated with each of our five knowledge models here. For example, if we look at Figure 13, we see that center for the topographic patch moves around the center of the region, within a fairly constrained radius. The strategy behind this knowledge model is to basically place individuals at the whole region so that overall intake of resources by all individuals for this strategy will be relatively constant. From this standpoint, it is an exploratory knowledge strategy that is sampling the current total environment.

Looking next at situational knowledge, Figure 14, we see a completely different scenario. This knowledge model is tracking the optimum. Notice that there are two lines connecting each of the dense regions. This corresponds to the fact that the maximum valued cone travels around the circuit in two trips, and that situational knowledge jumps quickly from its old location to its new one and then exploits the new one intensely, given the tight radius at each location. There is variability in that the cones, when moved to the new quadrant, are placed randomly within it (not always in the same spot). Situational knowledge is an example of an exploitive knowledge model. In our traveling band analogy, it corresponds to the avid fans that follow their favorite group from venue to venue.

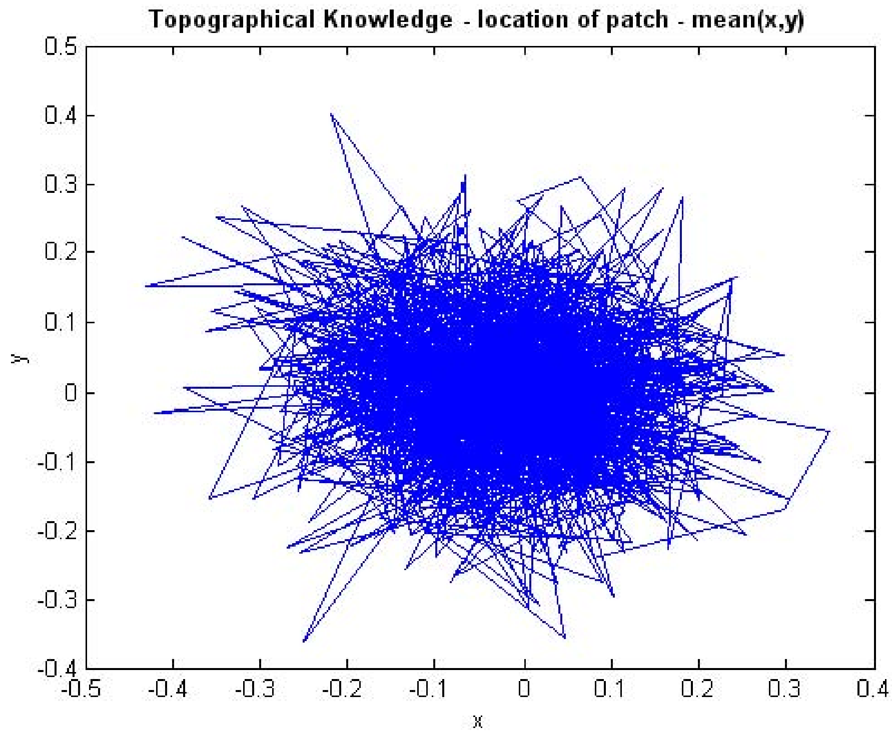

FIGURE 13 Change in the location of the center of the bounding box for topographic knowledge over 1,600 time-steps

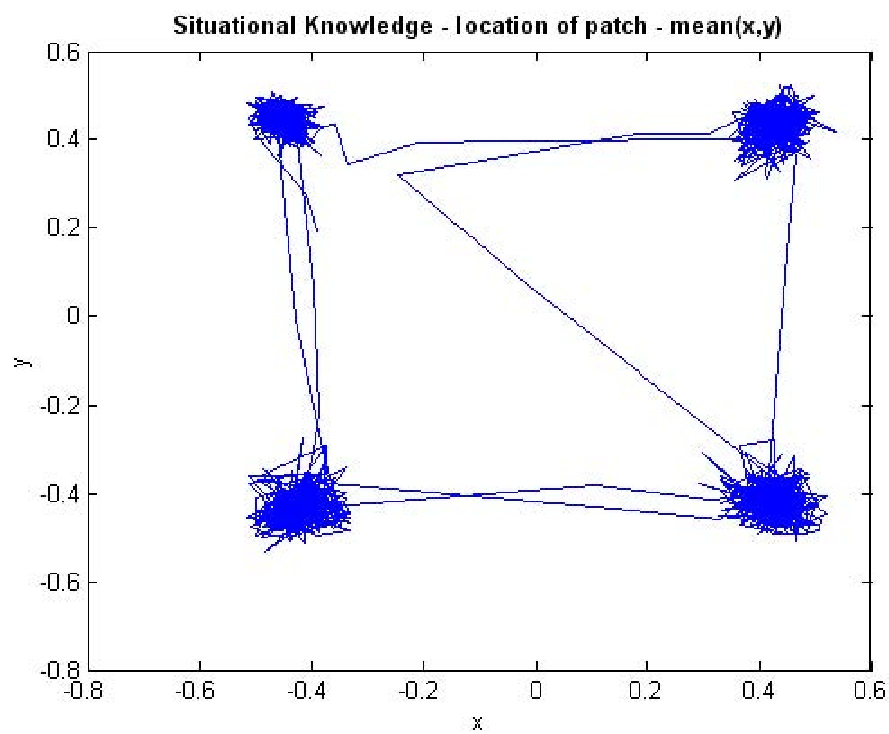

FIGURE 14 Change in the location of the patch center for situational knowledge over 1,600 time-steps

The movement of the domain knowledge patch is given in Figure 15. Domain knowledge was classified as an exploitative knowledge source in previous work, like situational knowledge. However, note that while it locates individuals on each of the four quadrants following the best one, there is an interesting emergent behavior here. There is a diagonal trace between the third and first quadrant. What this means is that this knowledge source is allowing those individuals to follow a shortcut from quadrant three back to quadrant one from where the best peak started. One reason for this is that given the popularity of the exploitation approach, there are many individuals attracted to the best peak by the time it gets to the third quadrant. The domain knowledge model uses the gradients to predict where the best cone will be going, and the individuals take a shortcut to get there ahead of the group.

This strategy is akin to those individuals who, when the number of exploiters becomes dense, are able to use their knowledge to project future venues and strive to be the first to arrive at the new opportunity. While this is an exploitive knowledge model, it has a relatively small population size, because if everyone were to take this choice, it would become disadvantageous. This works well when only a few are able to do it.

In Figure 16, we see the trajectory for normative knowledge. This knowledge was viewed as a type of exploratory knowledge source. Here we see that unlike topographic knowledge, it

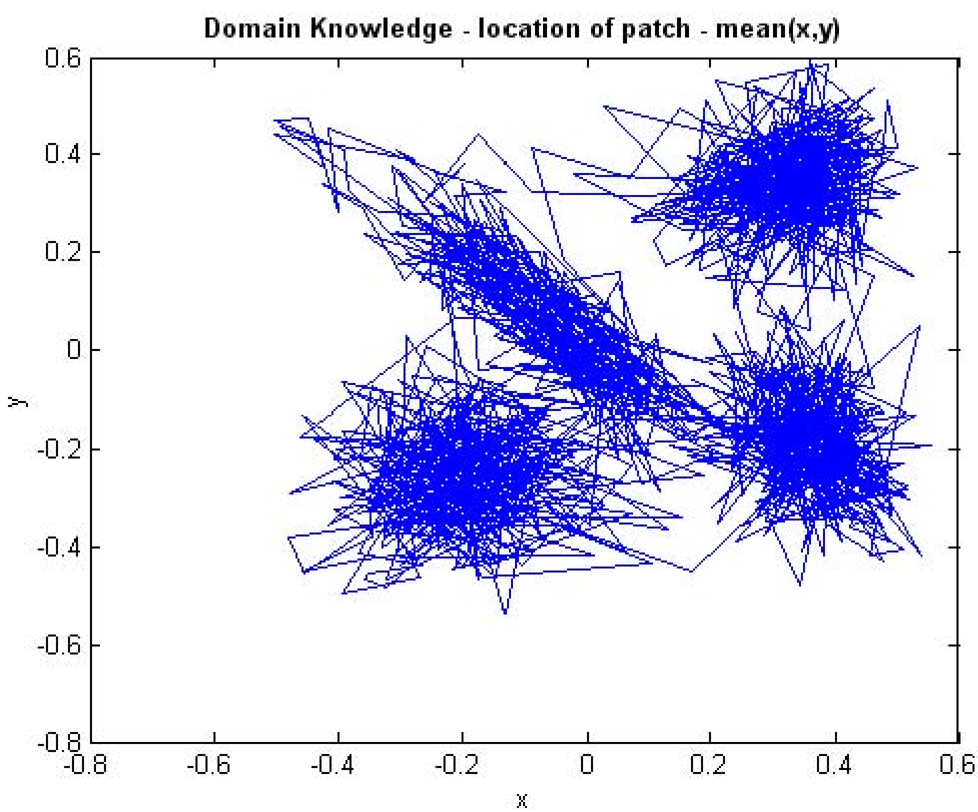

FIGURE 15 Change in location of the patch center for domain knowledge over 1,600 time-steps

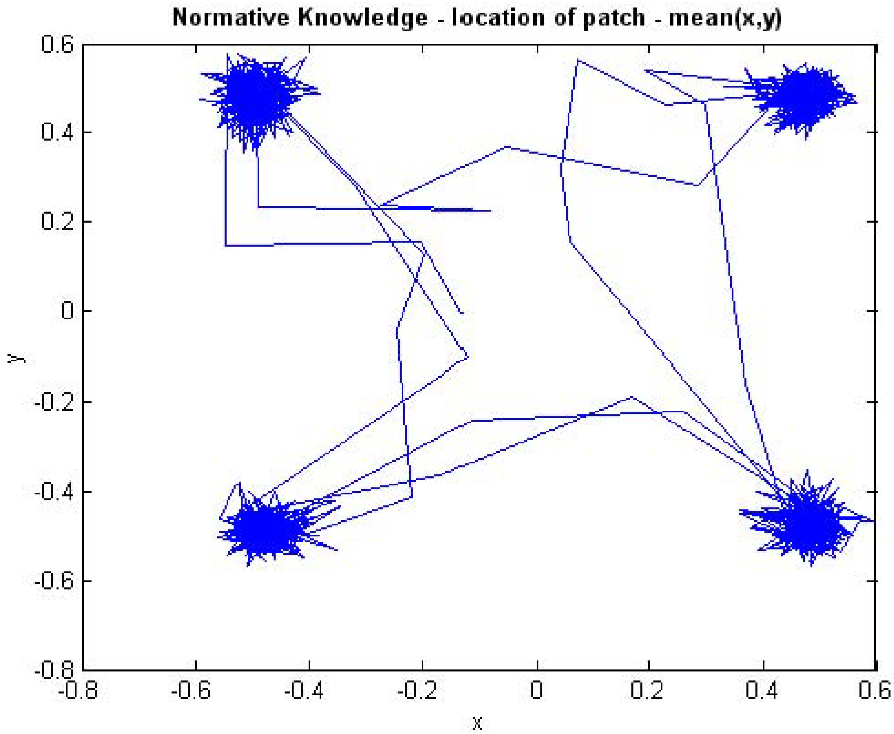

FIGURE 16 Change in location of the patch center for normative knowledge over 1,600 time-steps

moves from quadrant to quadrant following the optimal cone. Notice that again there are two connecting arcs between each quadrant, signifying the two cycles in our experiment. But now the arcs have a more meandering nature than that of the situational knowledge source. Whereas topographic knowledge explores the entire region, normative knowledge explores a subregion and is able to produce bounding boxes that connect one peak area to another. The key difference between normative knowledge and situational knowledge is that individuals using normative knowledge find and exploit the peak first, with situational knowledge directing individuals to follow. So while they both track the agents who follow their models to the best peak, they do it at different rates.

Finally, history knowledge is tracked in Figure 17. This knowledge model is focused on learning the pattern sequence. One can see that the radius of the patches is more spread out than that of situational and normative knowledge. Those using the history approach are able to encode trends and learn from them.

Other distinctions can be made between the knowledge models on the basis of parameters such as patch size, patch capacity, patch performance, and patch location. Looking first at the patch size in Figure 18 shows that topographic knowledge has the largest patch size, with normative and situational knowledge having the smallest sizes. All three patch sizes are relatively stable during the 1,600-year period. Both history and domain knowledge patch sizes experience an increase in size at the onset of a new cycle. Once the trajectory of the new cycle is determined, history knowledge returns to a stable patch size, similar to situational and normative knowledge. Domain knowledge, however, continues to decrease in patch size cycle, exploiting the cumulative changes in gradients.

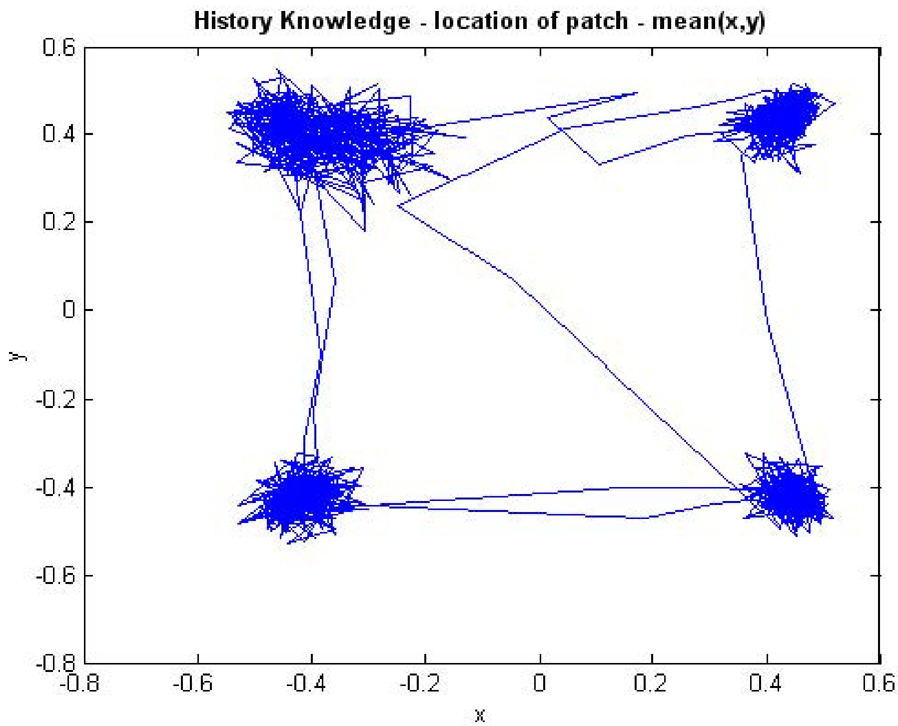

FIGURE 17 Change in location of the patch center for history knowledge over 1,600 time-steps

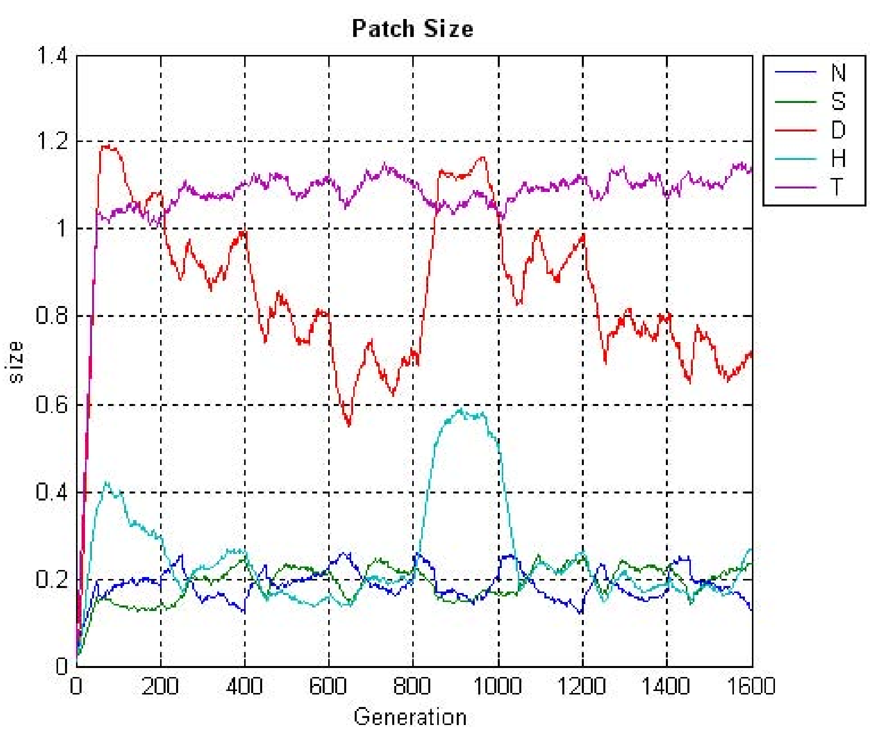

FIGURE 18 Patch size for all five knowledge sources over 1,600 time-steps

Patch capacity (Figure 19) corresponds to the number of individuals occupying each of the patches. Since the population is 200, and since not all individuals are found exactly within a patch, the total is somewhat less than 200. What is interesting here is at the onset of a cycle, history and domain knowledge recruit the most individuals, since they have information that can be used to predict the pattern of change. Topographic knowledge recruits a fairly constant number of individuals. However, situational and normative knowledge recruit more as the cycle continues, inheriting individuals from perhaps the domain and history models.

In Figure 20, patch performance is observed. What is interesting here is that each knowledge model exhibits a gain function that increases and then begins to decrease exponentially, but the adjustments take place at different frequencies. For example, topographic knowledge exhibits this shift at the onset of each cycle and is stable in between. The cycle for domain knowledge is longer (around 800 years) and shifted from the origin. Normative, situational, and history knowledge exhibit a higher frequency of change, around every 200 cycles. But the shifts are complementary in the sense that history and situational knowledge are going down, while normative is shifted so it is going up at the same time. This reflects that wave pattern of occupation mentioned earlier.

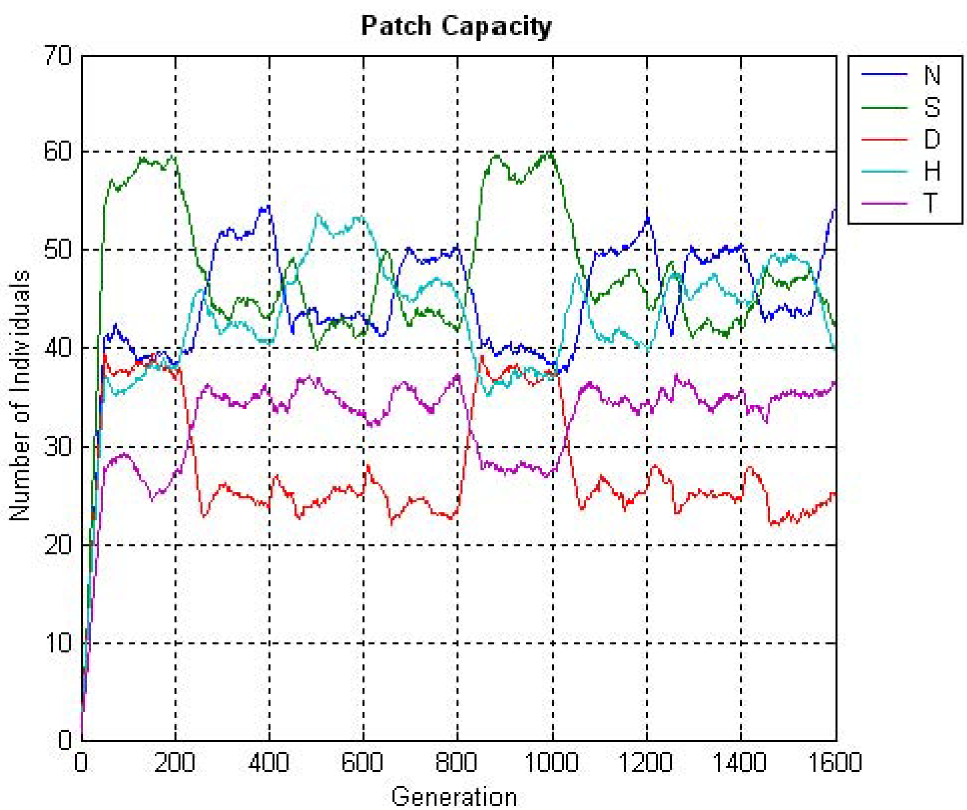

FIGURE 19 Patch capacity for all five knowledge sources over 1,600 time-steps

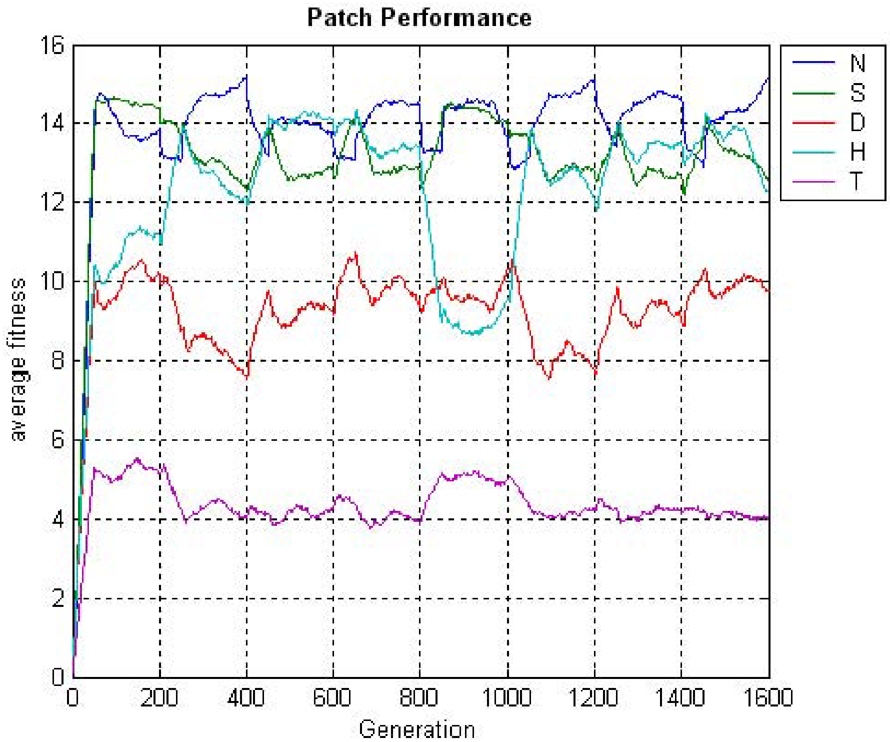

FIGURE 20 Patch performance for all five knowledge sources over 1,600 time-steps

In Figures 21 and 22, we observe the location of the patch centers in terms of the x and y coordinates, respectively. Notice that topographic knowledge is located consistently at the center of the region. However, we noticed earlier that its patch size increases and decreases over time, reflecting the need to produce more exploration. Situational and normative knowledge are quickly relocated into the center of the new quadrant at each phase. However, once they are in a quadrant, their patch size then changes, typically getting smaller. Domain knowledge, on the other hand, exhibits a hedging affect, which means its patch center tends to hedge back toward quadrant one as the cycle continues.

CONCLUSIONS

In this paper, we have investigated the impact of environmental dynamics on roles available to individuals. Here, the addition of a cyclical dynamic component to the model allowed each knowledge model to exploit a different aspect of the environmental dynamics. In previous reports that used a static configuration, certain knowledge sources, such as history and domain knowledge, had too little information to apply their expertise; therefore, those individuals that were controlled by those knowledge models exhibited behavior that was similar to other exploiter knowledge sources.

Here the additional information provided by the environmental dynamics allowed both history and domain knowledge to generate a pattern of patch movements, which was able to

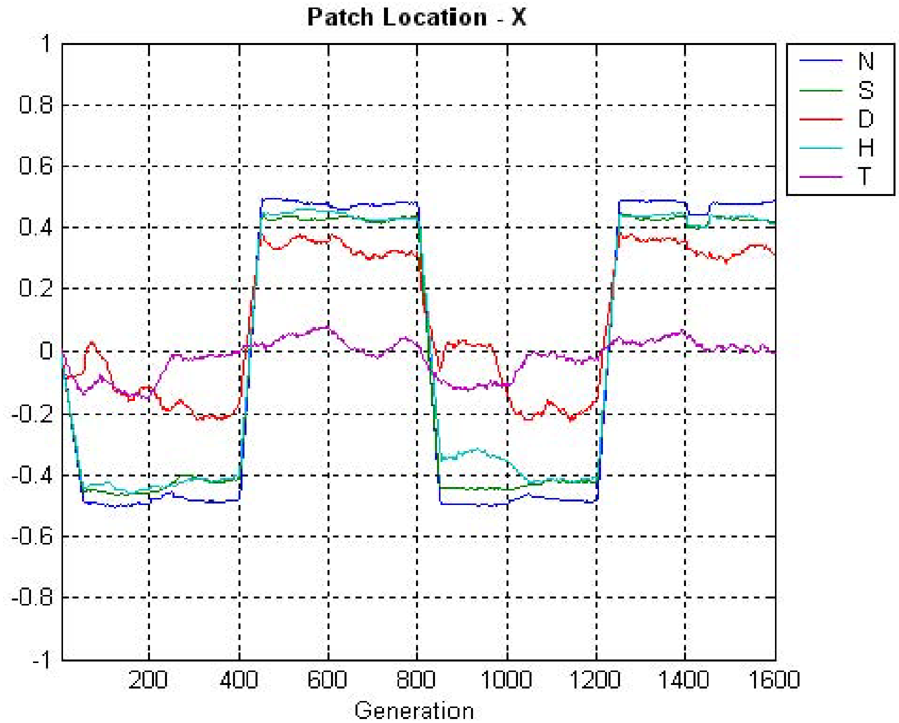

FIGURE 21 The x axis location of the patch center for all five knowledge sources over 1,600 time-steps

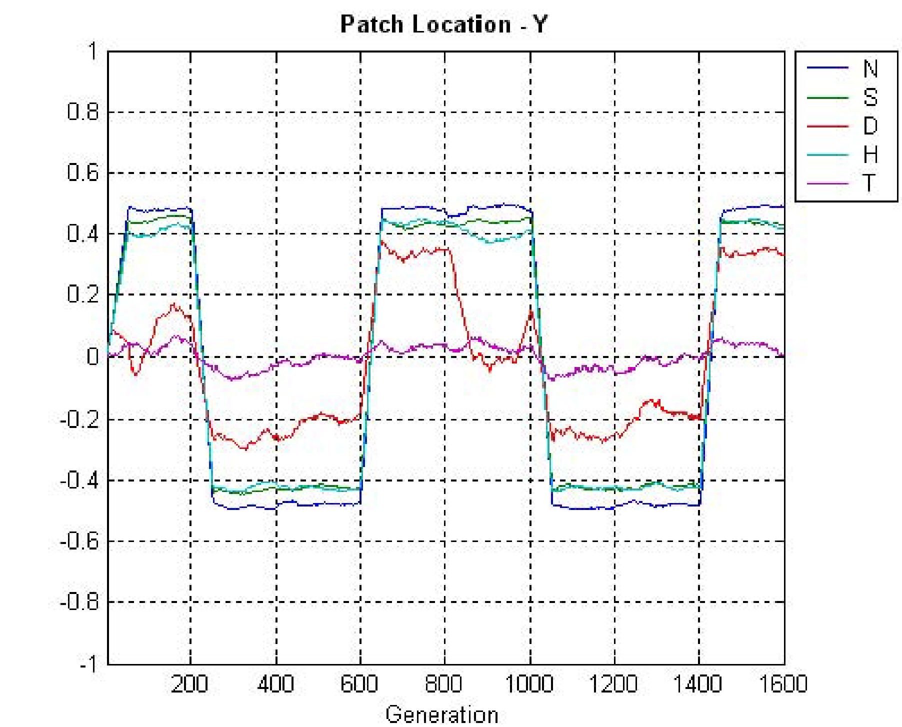

FIGURE 22 The y axis location of the patch center for all five knowledge sources over 1,600 time-steps

guarantee them some success with a small group of individuals. Thus, three distinct exploiter strategies emerged, when there was only one before. History and domain knowledge were able to predict aspects of the dynamics and use them in different ways. Both were able to arrive at the new optimal quadrant before situational knowledge, and both would move out as the agents driven by the situational model moved in. However, while history knowledge moved to the next patch, domain knowledge demonstrated the ability to hedge its path and move ahead to patches farther down the route. This behavior became clear as the number of individuals following a patch increased after year 600 and 1200.

Likewise, the cyclical nature of the environmental dynamics caused the two exploratory knowledge sources to also differentiate their behaviors. While topographic knowledge focused on a central location, normative knowledge mined the related regions but adjusted its patch location as more exploiters were attracted there. In future work, we anticipate that by adding in different dynamics, we will affect the mix of strategies that emerge.

REFERENCES

Barnard, C.J., and R.M. Sibly (1981). "Producers and Scroungers: A General Model and Its Application to Captive Flocks of House Sparrows." *Animal Behavior* 29:543–550.

Beckers, R., J.L. Deneubourg, and S. Goss (1992). "Trails and U-turns in the Selection of the Shortest Path by the Ant Lasius Niger." *Journal of Theoretical Biology* 159:397–415.

Charnov, E.L. (1976). "Optimal Foraging: the Marginal Value Theorem." *Theoretical Population Biology* 9:129–136.

Chung, C. (1997). *Knowledge-based Approaches to Self-adaptation in Cultural Algorithms,* Ph.D. Thesis, Wayne State University.

Chung, C., and G.R. Reynolds (1998). "CAEP: An Evolution-based Tool for Real-valued Function Optimization Using Cultural Algorithms." *International Journal on Artificial Intelligence Tools* 7(3):239–291.

Clayton, N.S., D.P. Griffiths, and A. Dickinson (2000). "Declarative and Episodic-like Memory in Animals: Personal Musings of a Scrub Jay," in *The Evolution of Cognition.* Edited by C. Heyes and L. Huber, MIT Press, Cambridge, MA.

Colorni, A., M. Dorigo, F. Maffioli, V. Maniezzo, G. Righini, and M. Trubian (1996). "Heuristics from Nature for Hard Combinatorial Optimization Problems." *International Transactions in Operational Research* 3(1):1–21.

Dorigo M., V. Maniezzo, and A. Colorni (1996). "Ant System: Optimization by a Colony of Cooperating Agents." *IEEE Transactions on Systems, Man, and Cybernetics* 26(1):29–41.

Epstein, J., and R. Axtell (1996). *Growing Artificial Societies.* MIT Press/Brookings Institute, Cambridge, MA.

Goss, S., S. Aron, J.L. Deneubourg, and J.M. Pasteels (1989). “Self-organized Shortcuts in the Argentine Ant.” *Naturwissenschaften* 76:579–581.

Holland, J.H. (1998). *Emergence.* Addison-Wesley Press, Reading, MA, pp. 1–10.

Hölldobler, B., and E.O. Wilson (1990). *The Ants.* Springer-Verlag, Berlin, Germany.

Hu, X., C.R. Eberhart, and Y. Shi (2003). “Engineering Optimization with Particle Swarm,” in *Proceedings of 2003 IEEE Swarm Intelligence Symposium,* Indiana, IEEE Press, pp. 53–57.

Iacoban, R., R.G. Reynolds, and J. Brewster (2003). “Cultural Swarms: Modeling the Impact of Culture on Social Interaction and Problem Solving, ” in *Proceedings of 2003 IEEE Swarm Intelligence Symposium,* Indiana, IEEE Press, pp. 205–211.

Jin, X., and R.G. Reynolds (1999). “Using Knowledge-based Evolutionary Computation to Solve Nonlinear Constraint Optimization Problems: A Cultural Algorithm Approach,” in *Proceedings of the 1999 Congress on Evolutionary Computation,* Piscataway, NJ, IEEE Service Center, pp. 1672–1678.

Kennedy, J., and R.C. Eberhart (1995). “Particle Swarm Optimization,” in *Proceedings of the IEEE International Conference on Neural Networks,* Perth, Australia, IEEE Service Center, pp. 12–13.

Kennedy, J. (1999). “Small Worlds and Mega-Minds: Effects of Neighborhood Topology on Particle Swarm Performance,” in *Proceedings of the 1999 IEEE Congress on Evolutionary Computation,* Piscataway, NJ, IEEE Service Center, pp. 22–31.

Kennedy, J., R.C. Eberhart, and Y. Shi (2001). *Swarm Intelligence.* Morgan Kaufmann Publishers, San Francisco, CA.

Maniezzo, V. (2000). *Ant Colony Optimization: An Overview.* Available at http://www3.csr.unibo.it/~maniezzo/didattica/Vienna/ACOintro.pdf. Accessed March 12, 2005.

Morrison, R., and K. De Jong (1999). “A Test Problem Generator for Non-stationary Environments,” in *Proceedings of the 1999 Congress on Evolutionary Computation,* Piscataway, NJ, IEEE Service Center, pp. 2047–2053.

Ostrowski, D.A., T. Tassier, M. Everson, and R.G. Reynolds (2002). “Using Cultural Algorithm to Evolve Strategies in Agent-based Models,” in *Proceedings of the 2002 IEEE World Congress on Computational Intelligence,* Wakiki, HI, pp. 2–7.

Peng, B. (2005). *Knowledge and Population Swarms in Cultural Algorithms for Dynamic Environments,* Ph.D. Thesis, Wayne State University.

Reynolds, R.G. (1978). “On Modeling the Evolution of Hunter-Gatherer Decision-making Systems,” *Geographical Analysis* 10(1):31–46.

Reynolds, R.G. (1986). "An Adaptive Computer Model of Plan Collection and Early Agriculture in the Eastern Valley of Oaxaca," in *Guila Naquitz: Archaic Foraging and Early Agriculture in Oaxaca, Mexico,* K.V. Flattery, Editor, Academic Press, pp. 439–500.

Reynolds, R.G. (1994). "An Introduction to Cultural Algorithms," in *Proceedings of the Third Annual Conference on Evolutionary Programming,* World Scientific Publishing, pp. 131–139.

Reynolds, R.G., and B. Peng (2005). "Knowledge Learning and Social Swarms in Cultural Systems." *Journal of Mathematical Sociology* 29:1–18.

Reynolds, R.G., and S.M. Saleem (2005). "The Impact of Environmental Dynamics on Cultural Emergence," in *Perspectives on Adaptions in Natural and Artificial Systems.* Oxford University Press, pp. 253–280.

Rychlyckyj, N., D. Ostrowski, G. Schleis, and R.G. Reynolds (2003). "Using Cultural Algorithms in Industry," in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium,* IEEE Press, pp. 187–192.

Saleem, S. (2001). *Knowledge-based Solution to Dynamic Optimization Problems Using Cultural Algorithms,* Ph.D. Thesis, Wayne State University.

Stephens, D.W., and J.R. Krebs (1986). *Foraging Theory.* Princeton University Press, Princeton, NJ.

Wynne, C.D. (2001). *Animal Cognition — The Mental Lives of Animals.* Palgrave Macmillan, Great Britain.